The Microcomputer Club of Melbourne, Inc

Correspondence to P.O. Box 60, Canterbury, Victoria, 3126

May 1987 Newsletter

Registered by Australia Post Publication No. VBH3601

Contents

Micom Information	- Page 2
Prez Sez	- Page 3
Sec's Say	- Page 3
Next Meeting Details	- Page 3
Librarian's Report	- Page 3
The C Programming Language	- Page 4
British 'Phone Fraud	- Page 5
Getting The Most From Your Hard Disk	- Page 6
Ethics of Computer Conferencing	- Page 7
The Great Chip Race	- Page 8
Micom's Bulk Buys	- Page 9
Micom's Special Interest Groups	- Page 9
Members' Messages	- Page 10
Event Calendar	- Page 11
Micom Discounters	- Page 11

Meeting Time and Place

The Microcomputer Club of Melbourne, Inc meets at 2pm on the third Saturday of each month at Victoria College, Burwood Campus, in the Community Resources Centre, Building E.

Next meeting

2pm, Saturday, May 20th

Production Credits

This edition was edited by Ralphe Neill and camera-ready artwork was produced by Greg Black on Avid Micro Wares' DEC LN03 Laser Printer. "Fancy-Font" borders and headings were produced by Peter Jetson on a Gemini 10x printer. The newsletter was printed by Lisa Rigby on Micom's printing press.

Copyright and Reprint Privileges

Articles herein that are copyrighted by individual authors or otherwise explicitly marked as having restricted reproduction rights may not be reprinted or copied without written permission from Micom and/or the authors. All other articles may be reprinted for any non-commercial purpose if accompanied by a credit line including the original author's name and the words "Reprinted from the Micom Newsletter, PO Box 60, Canterbury, 3126, Victoria".

Newsletter Contributions

Articles, papers, letters, drawings and cartoons are actively sought for publication in the Micom Newsletter. Contributions can be sent to: The Editor, Micom, PO Box 60, Canterbury, 3126. Articles can also be uploaded to the CBMS by prior arrangement with Peter Jetson or you can give your item to a committee member at any Micom meeting. It would be appreciated if contributions were submitted on disk as that means they don't have to be re-typed! With such a wide range of computers available amongst our many members we should be able to accept almost any type of disk, but CP/M or UNIX formats are certainly the easiest. Absolute deadline for articles is the weekend after the Micom meeting.

Membership and Subscriptions

At a recent committee meeting it was decided that it was unfair to charge people $15 when they join half-way through the year or later. So the following fee structure now applies.

If you join:

January through to end of May	$15
June through to end of September	$10
October to the end of the year	$5

To become a member of Micom send your cheque to:

Micom, PO Box 60, Canterbury, Victoria, 3126

Micom CBMS

Micom operates a CBMS (Computer Based Message System) for its members. The system is available 24 hours a day on (03) 762 5088 (two lines), using the following data format: 8 data bits, 1 stop bit, no parity, full duplex, CCITT V21 300 baud standard. Make sure your modem is set to ORIGINATE mode. Visitors have limited access to the message system only.

Your first time on the CBMS

Your username on the system is your first and last names separated by a space and spelt exactly as they appear on your newsletter's mailing label. If your label only contains an initial, you cannot be provided with CBMS access until we find out your full name. Your initial password is your first name. You can change your password to anything you like (and can remember!).

Micom Meetings

The Microcomputer Club of Melbourne (MICOM) meets at 2 pm on the third Saturday of each month. Meetings are held at Victoria College, Burwood Campus, in the Community Resources Centre (Building E).

Next meeting: Saturday, May 16, 1987 at 2 pm

Your Committee

President	Geoff Halprin	859-9871
Vice-President	Andrew Stewart	428-1290
Secretary	Eric Salter	861-9117
Treasurer	Stephen Welker	(059) 78-6133
Membership	Stephen Thomas	836-7364
Editor	Ralphe Neill	via-CBMS
SIG Co-ordinator	Brett Robertson	773-2213
Librarian	John Pryor	277-4140
CBMS Operator	Peter Jetson	762-1386
Purchasing	Stan Smith	570-7292
	Andrew Stewart	428-1290

Special Interest Groups

CP/M	Steve Welker	(059) 78-6133
MAGE	Ralphe Neill	via CBMS
Otrona	David Broadbent	528-2792
Prolog	Tim Falkiner	836-4023
Social Implications	Tony Smith	370-3566
Motorola	Frank Crivelli	434-3806
PCIG	Andrew Moore	596-1896
Lap-Top Group	Peter Gorring	62-5051 BH

Personal Computer Interest Group

Chairman	Andrew Moore	596-1896
Secretary	Tim Falkiner	836-4023
Treasurer		
Software	Sid Prendigast	580-7367
	John Dalstead	(054) 28-2642

Pesterable Micom People

These people are available to help members with technical problems they may have. Please keep calls inside reasonable hours.

Speciality	Name	Phone
Cromemco C-10	Tom Beer	817-2571
6809, Flex, Omegasoft Pascal	Bob Byrne	420-5702
68xx, Flex, Uniflex	Alan Fowler	857-7128
Dr Who, Star Trek, UNIX, BASIC	David Green	241-6673
Disk Drives/controllers	Robert Green	818-6343
Cromix, UNIX, C	Ralphe Neill	via-CBMS
Z-80 Assem, CP/M, Mex	Eric Salter	861-9117
MRM, Apples, Pascal, Assembler	Stephen Thomas	836-7364
TurboDOS, Aztec C, C-tree	Stephen Welker	059 78-6133

Advertising Rates

Display Advertising (from supplied artwork):

Full page	(max 27cm x 19cm)	$70.00
Half page	(max 13.5cm x 19cm)	$40.00
Quarter page	(max 13.5cm x 9.5cm)	$20.00
Business card size	(max 6cm x 9.5cm)	$5.00

Extra charges will apply for artwork preparation or finishing.

Note the new business card size advert now available. Create a simple advert by just sending us your business card (or design your own artwork to the maximum size shown).

A 10% discount applies on the above prices if you want to run the same advertisement in several successive issues.

Inserts: Advertiser pays for postage of newsletter.

Commercial classifieds: $1.00 per 80 column line.

Non-commercial members' messages: Free of charge.

PREZ SEZ

OK you lot, here's the monthly waffle!

If you missed the last meeting, you missed out on Douglas Hale's talk on static electricity, which was well received by all who were there. You also missed out on Marcus Wigan's demonstration of expert systems and how to drop names.

At the next meeting, we have something for just about everyone. There will be a few games machines demonstrating such things as 3-D breakout (remember your glasses!) and other impressive demonstrations. There will be a C-tutorial by Eric Salter, the illustrious author of our on-going series, and a short talk by Steve Welker on the system he's building in his spare time.

In coming months we hope to have a speaker on the copyright situtation in the computer industry, and we just might see the return of the Bob Pascoe.

At the time of writing this we don't have a stand at PC'87. However, if you are reading this newsletter after finding it on a stand at the PC show, a quick word about the club. We are the oldest (or second oldest by less than a week) club for micro-computer enthusiasts in the country. We are exactly that - a club for computer enthusiasts. People who don't just want to keep the cheque books balanced on the computer. People who actually want to have fun with them and learn more about them. We are not snobs, although our membership does boast some of the leading minds in the industry. We are a social group of people who get together once a month to exchange ideas and conversation. Our membership ranges from the aforementioned experts to complete novices who have been bitten by the bug (and are still trying to get rid of it). As you will be able to see from the rest of the newsletter, our activities are many and varied and we hope, by the fact that you are still reading, that you consider yourself an enthusiast and will come to our next meeting to see for yourself.

The current membership of MICOM Inc. is about 270 or so. This falling membership problem that we have incurred over the last few years has led us to do some soul-searching and we have decided to have a brainstorming session at the next committee meeting. ANYONE who is interested in the future of MICOM Inc. is welcome to attend. Either see me at the meeting, or ring me sometime for details.

There are a few suggestions in the wings and they will stay there unless the committee receives positive input from the membership. If you don't say anything, nothing will happen!

Suggestion 1: That the club starts a programming project - a conferencing system. Programming would be done in C, using the Unix model. The project would require resource management and anyone not pulling their weight would have to pulled from the project. If it goes ahead, the task would be divided into mangageable tasks that are well defined.

Suggestion 2: That the club holds a MICOM Expo. This is the same idea that fizzled last year because only a couple of people actually responded to cries for help [see above].

Hope to see you at the next meeting ... Gooff Halfepint

SEC'S SAY

Just a brief column this month as I face up to my last "10101" days at Hamilton.

At the last committee meeting we decided that the next two committee meetings would be turned over to foward planning, where we would sit down and look at MICOM's directions. All are welcome to attend! The next meeting in on Saturday May 23 at 2pm SHARP. If you wan't to come, please ring either Geoff Halprin or Eric Salter and we'll tell you where it is.

All committee members are advised that their presence is required at the May 23 meeting. All excuses for not being there will be disregarded with the exception of sudden death, when a written apology will be looked upon favourably!

One of the other things we discussed was beginning a conferencing system. We have a IBM clone so we could set up our conferencing system on a separate line to the message system and maybe devote the message system to up/downloading? We could use the OPUS software for conferencing and obtain a FidoNET number so we could interface worldwide.

The disadvantage is that we lose multi-user capability if we go to the IBM and leave the S100 system we have now.

We also discussed a MICOM programming project - to write a conferening system, like BIX in the States, from scratch using C as the programing language. It would require quite a few people who are skilled at C, but if we put in the effort and planned well, this could be another Munchkin project (back in the days of Philip Freidin).

I invite those who are interested in undertaking this project to contact us. MICOM is looking quite good at the moment with meetings being planned. Why not get involved?

Eric Salter

THE MICROCOMPUTER CLUB OF MELBOURNE INC.
May General Meeting

Saturday May 16th at 2 pm Victoria College
Building E, Burwood Highway, Burwood

What's on: PC Interest Group Meeting

The C Programming Tutorial #1
Introducing 'C'

Games Day
Generations of Computer Games

Introduction to the Microtrix System
using the Hitachi Z-80 chip

Refreshments

Beginners welcome - computer guru's encouraged to attend!

LIBRARIAN'S REPORT

Hi-Ho Silver-Oxide, the Loan ArRanger Writes Again. (sic)

I thank those who have donated to the Library over the last few months and, once again, I request donations of things. "Things" includes old newletters (any that you no longer require), books and magazines (particularly Lifelines, ETI and EA).

Also, those of you who may be in the employ of organisations such as the SEC, Telecom, BHP or even the Department of Administrative Affairs (after all, we are a bunch of hackers) might be able to obtain from the Technical Library of your organisation any obsolete copies of D.A.T.A. Books or the I.C. Master (for that matter anything covering electronics). If you can help, I'd be more than willing to accept such material.

There are some overdue books and magazines out there. Please return them at the next meeting or post them to PO Box 60, Canterbury.

John Pryor

Contact

Nemostar Computing

for

Computer paper, labels, disks,
printer ribbons, and tapes.

Tel: Steve Welker (059)786-133

THE C PROGRAMMING TUTORIAL

by Eric Salter

This month, we begin looking at the structure of a C program. We will also look at some of the statements that are available in the language and we will see how you can "roll your own" using functions.

At the end of this, you should be able to, at least, write simple C programs. The full power of C does not happen until you begin to use data structures and pointers, but these are difficult topics and require a considerable amount of explanation - next month!

The Structure of a C Program

A C program is simply a text file that you create with a text editor or word-processor and that's all. You cannot run it directly because it has to be compiled. This means that a program, called a C compiler, must read your text file and convert each C statement into the binary code that your particular CPU understands and an execute directly. This means that your C program will run as a machine language program - fast - unlike BASIC statements that have to be interpreted each time the BASIC interpreter comes across them.

The other advantage is that a C program stands alone. It (usually) doesn't need any "run time" package to be in memory with it. A BASIC program needs the BASIC interpreter to make it do useful things. C programs, therefore, (usually) require less memory to achieve the same result as an interpreted program.

The C program consists of several sections which are part of every C program:

```
headers, definitions and macros

global variable declarations

main(argument_list)
argument declarations
{
        local variable declarations

        statements
}

function1(argument_list)
argument declarations
{
        local variable declarations

        statements
}

function2(argument_list)
.
.

function3 (argument_list)
.
.
```

A feature of a complete "runnable" C program is that it has the main() function. It is not actually necessary for C code to have a main() function before it is compiled. The necessity comes at "link" time when the executable code is created - the main() function tells the linker where to begin execution of the program. We can compile functions independently of the program and, indeed, of other functions and these can then be inserted at link time. In fact, many of the statements in a C program are just that - references to pre-compiled functions that reside in function libraries and are linked in to your code by the linker. It does this by searching through one or more libraries that you specify to the linker and, as it finds them, it will incorporate that code into yours. Any unresolved references to functions will cause the linker to issue a warning to that effect and you can rest assured that your code will crash!

Headers, Definitions and Macros

Anything written in this section is not part of the C language. 'Statements' here are called compiler directives and belong to the compiler. They are never executed but are acted upon by the compiler when the text file is read in. They are preceded by the "#" character.

#include filename

The '#include' directive causes the file listed after the 'include' to be searched for, brought in and inserted into the input stream of the compiler. The most common use for this directive is to bring in standard headers that are used by all programs and contain definitions of system constants and functions related to file access. The names of these files are system dependent but some fairly constant ones are:

```
#include "stdio.h"  - I/O macro definitions
#include "ctype.h"  - character testing macros/type defs
#include "math.h"   - on UNIX systems define special math
                      functions
```

You will note that the name of the file you wish to bring in is enclosed in double quotes. This causes the compiler to search for the named file on the current drive or in the current directory (depending upon your operating system) unless a pathname is included within the double quotes.

If, instead, we use angle brackets <> to enclose the name the compiler/linker will search for the file in a predetermined directory that contains standard header files. Under UNIX this would be the '/usr/include' directory while on the Amiga, for example, it would be the 'include/' directory.

#define

The '#define' directive allows us to make extensions to C by defining macros. A macro allows us to use a shorthand way of expressing constructs that convey more meaning as to what is actually done than would the construct itself. It also may save keystrokes!

The simplest form of macro is the constant definition - for example:

```
#define MAXLINENUMBER  66
```

From now on, refererences to MAXLINENUMBER is equivalent to decimal 66. Not only does this convey more meaning than a raw '66', but it allows us to change the constant at one place and have it changed throughout the program should we need to at a later date.

The other use of macros is making simple from complex:

```
#define  max(x, y) ((x) > (y) ? (x) : (y))
```

Here, we have defined the macro max(x, y) with parameters. With each use of max(x, y) in our program the compiler will substitute the code x > y ? x : y (the ternary operator - see previous article for an explanation). This makes your code not only simpler but more easily understandable.

In one issue of BYTE magazine, there was an article describing a way to vandalise C by re-defining '{' as BEGIN and ';}' as END to make it look like PASCAL code. If they knew anything about C, they should have known that C is to PASCAL as an asprin is to a headache. Please, don't make C harder to read than it is (yes, this is an admission, it can be hard to read, but if written properly it is simply elegant).

Please note that a macro, such as the one above, is not a function call! It will be expanded into code, "in-line". Please be careful, therefore, as expressions passed to macros may be evaluated twice. This may become the source of program bugs if you pass an expression with function calls or ++ type operators (it will increment for each occurance of the operator)!

Functions

Functions are a part of all C programs. The main() function is also function - the place where execution commences. It is possible to have only the main() function but it is good practice to create functions which perform specific tasks that are called from main() and, indeed, from other

functions.

Once a function is written and debugged, it doesn't have to be re-compiled. This saves us re-inventing the wheel and C programmers develop their own libraries of functions that they have created over the years and use in their everyday code as if they were part of the language.

The structure of a function is:

```
type function_name(argument_list)
argument declarations

{

        declarations and statements usually
        including a 'return()'

}
```

For example, the max(x, y) we defined as a macro, could also be written as a function:

```
int max(x, y)
int x, y;

{

        x > y ? return(x) : return(y);

}
```

Here we have defined our function called 'max' to be a function which will return an integer. It has the formal parameters 'x' and 'y', which are defined as integers. The body of our function is the ternary operator as before, except that the statement 'return()' is passing control and an integer value back to the calling function.

We can see that a function can have multiple terminations. For purposes of explanation the function could also have been written as follows:

```
int max(x, y)
int x, y;

{

        if (x > y)
                return(x);

        else
                return(y);

}
```

Note also that we have defined the two arguments. We must do this as these are not the same variables that would be used when we call this function even though they do, of course, have the same value. They are, in fact, local to the function. Whatever we do to these variables cannot affect the real variables in the calling routine. We spoke about this in a previous article when we talked about the scope of variables as being one of the attributes of a variable.

Call By Value vs Call By Reference

In the above example when we call the function, probably from main(), as in:

```
max(new_value, old_value);
```

the variables 'new_value' and 'old_value' are the actual parameters we use. These variables are passed to the function by value. This means that the actual values of the two variables are copied into 'x' and 'y'. We never actually compare the variables new_value and old_value. x and y are local to the function and have no scope outside its call and are destroyed (deallocated on the stack) upon return.

For a function to effect a change in the variables outside itself, that variable must either be an external variable, i.e. one with global scope, or the variable can be passed to the function by reference. In call by reference, the address of the variable in memory is passed and this is used by the function to access the real variable. The address of the variable is a pointer to that variable.

Pointers give C great power but they can be dangerous in functions if you don't remember that you are operating under call by reference. You may destroy the data you are operating on. Note that functions themselves have global scope for a given compilation unit so that any function can call another. Functions can also call themselves i.e. recursion!

Next month, we look at pointers and arrays. We will spend some time on pointers as these form the basis of the power of C. We can have pointers to variables, pointers to arrays, pointers to arrays of pointers, pointers to structures, pointers to functions etc ...

EXTRA! British Telecom Pay Phone Phonecard Broken?

by Peter G. Neumann

Britain is currently just at the tip of an iceberg regarding an apparent vulnerability in its debit cards for British Telecom pay phones. The debit cards can be purchased from all sorts of shops, and come in a range of denominations such as 5, 10, 40, or 100 calling units. The system has been in use for a year or two, and card pay phones are both widely accessible and very popular. (If you've ever tried to use coins in a London call box, you know that it is quite an experience.)

My best guess is that it has a holographic stripe, and that a destructive write is used effectively to burn out a part of the hologram corresponding to each message unit - making it difficult to ADD units to the card.

Unfortunately, a relatively simple doctoring of the card has been discovered that threatens the whole scheme and makes a card indefinitely reusable (at least until the system is either modified or withdrawn).

An article appeared as the front-page lead story in The Sunday Post (West Scotland) with the banner headline "DIAL WORLD WIDE FOR NOTHING - TELECOM HIT BY 'PHONE FRAUD'".

The article noted that the trick was discovered by a British soldier "fed up with paying a fortune to call his Scottish girlfriend". The word is now spreading around British troops and can be expected to be widely known in a very short time. (The newspaper states that they know how it is done, and have proved that it works. It cites a variety of calls that they were able to make without any debit to their card.)

The consequences of the propagation of this trick are awesome to contemplate.

The system was presumably billed as "foolproof". But "foolproof" is not good enough against intelligence - although it should be pointed out that the card is not a smart-card in the usual sense. There is no user identification number required and no use of encryption. The AT&T credit card number seems somewhat safer as it is quickly revocable on an individual basis. On the other hand, the convenience of the BT phone card is certainly appealing.

Contact

Nemostar Computing

for

CP/M, TurboDOS, Z-systems
hardware and software systems.

Tel: Steve Welker (059)786-133

Taking Advantage of Your PC's Hard Disk
Part I, File Management

by Andrew Moore

based on an article by
Barry Gordon
New York Personal Computer, Inc.

This article is about organising files on any hard disk attached to a member of the IBM-PC family, including the PC with an Expansion Unit, and certainly the PC AT. The following suggestions have evolved from several months of using the IBM-PC XT as a follow-on system to the IBM-PC. I hope these thoughts prove useful to those who work with hard disks on the IBM-PC - particularly those who are new to it.

File Management

The hard disk is not merely an overgrown diskette. It can be used that way, but you would be inviting serious file management problems in doing so. The hard disk has other capabilities you should utilise and there is no reason not to take full advantage of the hard disk's potential.

About the last thing you want to deal with is a ten-megabyte hard disk containing all its files in one directory. (There is a limit of how many files the root directory can contain, so the disk may give you an error message even though much of the disk is not filled.) With intelligent organisation, the hard disk retains its speed advantage over the diskette and you are better able to keep track of your files as well. The name of each file should use filename and the three-letter extension. Sooner or later you will need all the help you can get in remembering what each file contains. Most importantly, though, your files should be organised among several directories.

The Root Directory

Each DOS volume (diskette or hard disk) has a root directory which DOS creates when it formats the disk. The root directory on your hard disk should contain a minimum of files, reserving the space for sub-directory names.

Nonetheless, a few files are essential in the root directory. When you format the hard disk, use the /S parameter: FORMAT C:/S which copies three .COM files to your hard disk:

```
(IBMBIO.COM)
(IBMDOS.COM)
COMMAND.COM
```

Notice that the first two are hidden files. You won't see them in any DIRectory listing, but the CHKDSK command will tell you they are there.

You should store the various DOS external command files and all data files in other sub-directories.

Sub-directories for Executable Files

Now let's consider locating your various executable program files - the .EXE, .COM, and .BAT files. You'll want to separate related programs into sub-directories of their own. For example, you may want one sub-directory containing all of the external DOS commands, another sub-directory with your word processing .COM and .EXE files, another with your accounting .COM and .EXE files, etc. until you have stored all of your executable programs into sub-directories. They are much easier to remember this way.

The root directory has no user-assigned name, but all sub-directories are specifically created and named using the MKDIR command, MD for short:

```
MD anyname
```

You can create as many levels of sub-directories as you like, but generally, the fewer levels you have to create, the simpler your directory structure will be. There will be times when a second or third-level sub-directory is needed (a directory created within another sub-directory), but multi-leveled sub-directories create a more complex path structure for both you and DOS to sort through.

Directory Sizes

You can fill a sub-directory with as many bytes as the disk will allow, but it is often convenient to limit the size of those sub-directories into which you regularly store data to the capacity of a single diskette, 320-360 Kb. This allows you to back up a directory using the COPY command:

```
COPY C:*.* A:
```

The COPYable directory size offers an alternative to the BACKUP and RESTORE commands, an alternative that many find easier to use. It offers portability as well as backup. If you want to verify the copy with its original, you can follow the COPY command with this:

```
COMP C: A:
```

However, if you feel confident using the BACKUP and RESTORE commands, the contents of a single sub-directory can be as many bytes as you like. The bigger it is, the more time-consuming the backup process becomes.

Another suggestion is to give each backup diskette a volume label. This is especially useful if you use the COPY command to back up your sub-directories. By using the /V parameter when you format the diskette, you can label your diskette with the same name as the directory it backs up.

It is rarely necessary to back up the entire hard disk. Typically, you store data changes in the files of a few sub-directories, those few being the only sub-directories you need to back up regularly. Many sub-directories never change and need backing up only once.

CP/M USERS GROUP

* News & Discussions
* Public Domain Software
  - SIG/M
  - CUG
  - CP/MUG
  - Software Tools RCP/M
* Byte Listings
* Disk Copying Available
  - 8" & 5.25"
* Refreshments
* Start time 7:30pm.
* 4th Tuesday in Month
* Community Resources Centre,
  Building E,
  Victoria College,
  Burwood Campus.
* Contact: Steve Welker
  (059) 786-133

THE ETHICS OF COMPUTER CONFERENCING

By John R. Cook
North Dakota State University

The objectives of the present study (which I have been carrying out into the ethics of computer conferencing) were twofold. First, an attempt was made to assess the extent to which some traditional moral principles governing face-to-face communication have gained acceptance by the users of computer conferences. This was done by assessing the prevailing community standards for computer conferencing on Compuserve's CB and on BITNET's Relay. The assessment was carried out using email and other message facilities on both of these networks to distribute a 24-item questionnaire entitled Questionnaire for Users of "CB's" and Other Computer "Chats". The second objective of the study was to explore some approaches that might be taken towards distributing questionnaires and collecting data over computer networks.

The 25 CB and 112 Relay users participating in the study tended to be male, undergraduate students in their mid-twenties and their average level of education was that of a college sophomore. Six of the CB users also turned out to be Relay users. In terms of their use of electronic journal and mail services, the largest portion of Relay users (21%) were Psychnet subscribers, followed by NetNorth (17%) and COMSERVE (16%).

CB users were found to be much more reluctant to participate in the study than Relay users, as evidenced by their lower response rate. Also, Relay users were found to be more accepting of a proposed rule requiring users to register their true names and identities before signing onto the system. Otherwise, there was a high degree of similarity in the responses of CB and Relay users on this survey.

The moral or ethical standards for computer conferencing revealed by the survey were found to be largely consistent with the moral principles governing face-to-face communication. One difference between computer-mediated and face-to-face communication was suggested by the conference users' greater acceptance of practices such as adopting nicknames that are considered sexually provocative or sharing details of one's intimate personal relationships on a public channel. However, when CB and Relay users were asked to evaluate the potential benefits of computer conferencing, they appeared to discount the importance of some of these practices, in a way that brought them back in line with traditional moral principles governing face-to-face communication.

In conclusion, it seems we do tend to attach importance to the same sorts of activities both on and off-line. In this respect, several traditional moral principles governing face-to-face communication have been found to be accepted by the users of computer conferences. One area where discrepancies between our on and off-line ethical standards do tend to occur, is in the larger range of behaviors we are willing to consider as morally acceptable in other conference users.

This is an abstract of the full paper, which is still in preparation.

EMAIL MISUNDERSTANDINGS

by taylor@hplabsc.HP.COM
University of California, Berkeley

Have you ever sent a computer msg that was misunderstood by the receiver and caused him/her to have ill feelings toward you that were never intended by you? Have you ever been the receiver in such a situation?

Because computer mail is becoming more and more a part of people's lives, especially in the academic and business domains, these kinds of problems in relationships will occur more and more frequently, it seems.

Computer mail makes written communication very easy, but this great ease may lead to problems in relationships due to misunderstandings of content, or more frequently, misunderstandings of motivation and feelings. More written msgs will be sent more quickly than without computer mail, so the chance of misunderstandings is greatly increased.

More specifically, the sender of a written msg may not have a very accurate idea of the way the receiver will understand the msg and of the impact on the receiver's feelings. And if the msg unexpectedly leads to misunderstanding or ill feelings, the sender is not present to correct the msg for the receiver, as he would be in oral communication by phone or in person.

On the other hand, the more frequent the two people interact orally, the less chance there is of misunderstanding a written msg, and the more chance there is of correcting a misunderstanding that did occur.

Several aspects and conjectures concerning written communication that one might consider include these:

1. Communication of Emotion
   a. Sender
      The sender may tend to communicate stronger emotion, than in oral conversation, because the receiver is not present. This strengthening may occur because the sender doesn't care as much about the effect on the absent receiver, or because the sender may unconsciously feel like greater emphasis is needed because he can't see the receiver's ongoing response and doesn't realize how strong it really is.
   b. Receiver
      The receiver may tend to feel stronger emotion than the sender intended to communicate, especially if the emotion is negative.
   c. Combined Effect
      Putting the two effects together, causes a written msg to generate a much stronger emotional impact than intended or expected by the sender. I think that a good example is when a person replies to a msg addressed to a bulletin board or an alias and accidentally sends his reply to the bulletin board or alias. He may then receive a barrage of msgs, some worded very strongly and negatively. The impact of each such msg is probably much greater than the sender intends or expects, and the end result may be very ill feelings.

2. Communication of subtle humor
   Subtleties may not come through the written medium as expected, leaving the receiver with different feelings than intended, maybe negative feelings.

3. Use of receiver's or sender's name
   Omission of either may leave the receiver with a feeling of very impersonal communication, not intended by the sender.

4. Delay or absence of reply
   Since sending computer mail is so easy, delay or absence of a reply may leave the sender wondering if he accidentally offended the receiver, especially if the msg send is short and the reply is expected to be short.
   Just the same:
   Since sending computer mail is so easy, some people may receive too many msgs to handle, and would be spending too much time answering mail if they tried to answer t be more true of managers and instructors, whom a lot of

The ZPC is now available at

Nemostar Computing

* Z-systems based (CP/M compatible)
* Hitachi HD64180 6/9MHz processor
* Individual components available
* Bankcard and VISA welcome

Tel: Steve Welker (059) 786-133

people have reason to contact. Maybe it would be good for the mail system to have form replies for this situation - at the user's request, a reply might be sent saying, "Due to the large number of msgs I receive, I am too busy to read and reply to your msg. Please come and see me during office hours if you would like to discuss the matter with me and receive an answer." Another reply might be, "I received your msg but I am very busy and will not be able to reply for some time."

Again, let me mention that the greater the frequency of oral contact between the sender and receiver the less the chance of these misunderstandings.

Since computer mail is becoming more and more important to our society, and to our relationships with other people, maybe its impact is worth researching by social psychologists.

IS THERE MORE ROOM FOR ERROR IN EMAIL THAN IN WRITTEN MAIL?

by Tom Mandel
Whole Earth 'Lectronic Link, Sausalito, CA

In principle, probably not much. In principle, both are exchanges of words about specific matters, i.e., written communications. Letter and memo writers have to learn the arts of letter and memo writing, the former which is taught in grade school and the latter which is usually a skill acquired on the job at work.

Not surprisingly, email writing is learned on-the-job too.

The ONE crucial difference between written and emailed writing may be the immediacy of writing email ON-LINE (this doesn't apply to off-line email writers). I find that I have a tendency to write email on line rather quickly, edit it fairly sloppily, and then I hit CR or "." or whatever the sending signal in my system is, and it's gone, forever carved in silicon chips and on magnetic tape. The ease of transmission and the lack of incentive to edit and rewrite carefully may be the only crucial difference between email and written mail in terms of skill requirements.

However, my remarks about email DO NOT apply to writing on bulletin board or computer conferencing systems, which is another matter entirely (although it has the same "immediacy" problem as email). Email is one-to-one or one-to-a-select-group - BBS stuff is much more like broadcasting ...

THE GREAT CHIP RACE

64-Megabit DRAM On The Way

Nippon Telegraph & Telephone is working on a 64-megabit DRAM and has already had some success with a 16-megabit DRAM.

The company will spend about US$300 million to build two laboratories for research on the 64-megabit chip. And teams at the labs will also work on chips with a capacity of 256 megabits or more.

However, many industry experts believe it will be several years before even 16-megabit DRAMs can be mass produced.

Engineers at Texas Instruments reacted skeptically claiming that on a 1-megabit chip, transistors are no further apart than 1 micron - the length a human fingernail grows in one hour.

IBM announced recently it is producing a 4-megabit chip and industry analysts said it "would be very surprising if IBM isn't working on 16-megabit and above technology."

Zilog 16-Bit CPU Is Programme-Compatible With Z80

Zilog introduces a 16-bit CMOS microprocessor that is fully programme-compatible with the 8-bit Z80 CPU.

The new processor, known as the Z280, will combine several common microprocessor functions including a memory management unit, on-chip caching, four DMA channels, UART, clock, programmable DRAM refresh and three counter/timers.

The Z280, which will have a 16-bit ALU and a Z80-compatible register set, can be programmed as an 8-bit Z80-compatible data path or as a Z-BUS 16-bit data path.

There will be two Z280 modes - system mode, for operating system code, and user mode, for application programs. There are separate stack pointers for each mode and protected system and user mode memory address spaces can be programmed on the MMU.

The chip also has a programmable multiprocessing mode that allows physical memory address space to be separated into local and global memory areas.

The Z280 will run at 10MHz, but versions with faster clock speeds will be ready within a year, according to Zilog. The Z280 was expected to be available now with introductory pricing in the US set at $27 in lots of 1000.

AMD 17-MIPS 32-Bit Microprocessor

Advanced Micro Devices will build a 32-bit CMOS microprocessor. The company claims it will be up to five times faster than any 32-bit device currently available.

The microprocessor, the Am29000, will have a clock frequency of 25MHz and operate at a sustained performance of 17 million instructions per second with a claimed peak performance of 25 MIPS.

The Am29000 will feature 192 general-purpose registers, 4 gigabytes of virtual address space with demand paging, concurrent instruction and data accesses, floating-point acceleration, a 64-entry on-board memory-management unit, and 40-ns cycle time. The microprocessor has a three-bus architecture consisting of separate 32-bit instruction, data, and address buses.

The company claims that several hardware vendors have already made commitments to develop products based on the chip. Officials predicted that the 29000 would be used in embedded controller environments that demanded high performance such as engineering workstations, laser printers, network protocol controllers, and robotics.

There could also be a VME card and a 29000-based IBM PC add-in card for UNIX applications.

NatSemi 32-Bit CPU

National Semiconductor will release the next generation of its 32-bit 32000 CPU family later this year.

The new microprocessor, the NS32532, will initially run at an average execution rate of 8 to 10 million instructions per second. It will also feature an integrated demand-paged virtual memory management unit, a four-stage overlapping instruction execution pipeline, on-chip caching with a 512-byte direct-mapped instruction cache and 1024-byte two-way set associative data cache, and an on-chip timing control unit. It will eventually run at 30MHz.

The 32532 allows for an internal memory bandwidth of 120 megabytes per second and an external bandwidth of 96 megabytes per second, although caching enables effective bandwidth of up to 240 megabytes per second.

NatSemi claims the 32532 outperforms both Intel's 80386 and Motorola's 68030 in clock rate and instruction cache.

Static 80C86, 80C88

Oki Semiconductor has developed static versions of Intel's 80C86 and 80C88 microprocessors.

The new chips will be available with a choice of top operating speed - 5 or 8MHz. But, unlike the dynamic versions, the static devices can run as slow as 0MHz - the older chips can run no slower than 2MHz.

With the static chips, the user can stop the clock for single-stop or standby requirements without losing data. According to Oki, the static processor is well suited for portable units such as instrumentation devices and laptop computers.

PURCHASING REPORT

Bargain of the month is available from Brett Robertson at Daneva - a 20-Megabyte hard-disk with controller for $900 or $630 without controller.

And Stan Smith has arranged for you to be able to buy a 20-Megabyte Tandon hard-disk with controller for $800.

Both of the disks are for IBM-PCs, of course.

If you need labels, we can offer stick-on labels very cheaply - 960 labels on tractor-feed backing for $5!

We still have some of that very high quality 70 GSM Superfine Bond paper with invisible perforations. When you rip the tractors off, it is A4 size paper that you can't tell that it has been through a computer printer.

The reason it is cheap at $30.00 for 2500 sheets is that the boxes are damaged but the paper is fine.

We still have two modems in stock. The first is an Interlink 12/75 modem that is manual-dial/manual-answer for $230.00. It supports V21 (300 baud full duplex), BELL 103 (US 300 baud full duplex) and V23 (1200/75 baud VIATEL standard). The modem is complete with power pack, telephone jack, push button handset and VIATEL software for the IBM PC or compatible.

The second modem is an Interlink A-12/12-X modem that is auto-answer/auto-dial $550.00. It supports V21 (300 baud full duplex) and V22 (1200 baud full duplex). The package is sealed but I think Stan said it was Hayes Compatible.

I still have about 50 single-density disks left in my stock for $1.50 each. I have mentioned to people that I don't think they are great disks so please use them as storage of your public domain software. Steve Welker has had three duds out of three boxes. Those disks were replaced.

There are also about 300 hard-sectored disks, 32 sectors per track. I know some Wang wordprocessors use these disks, and I've been told that some Hewlett Packard machines also use them. Ive sold about 30 of these disks at $1.00 each to people with these machines.

We also have 80 ribbons for Wang Daisy Wheel printers. I'm try to get $5.00 each. They are multistrike carbon ribbons of very high quality.

We have some more of the 60 x 5.25 inch diskette storage boxes for $16.00 each. 3.5-inch diskette boxes are very cheap at the Amiga User Group. We do not normally carry 8-inch disk boxes but I may be getting some for a good customer. Anyone else interested?

Current Disk Stock

Brand	Size	Sides	Density	Price (ea)
Interface	5.25	Double	Double	$1.60
Nashua	5.25	Single	Double	$1.70
Nashua	5.25	Double	Double	$2.00
Nashua	5.25	Double	Quad (96tpi)	$2.50
Nashua	5.25	Double	High (1.6MB)	$4.50
Nashua	3.5	Single	Double	$3.50
Nashua	3.5	Double	Double	$4.50
Nashua	8	Single	Double	$5.00
Nashua	8	Double	Double	$5.00
Verbatim	8	Single	Single	$1.50

Andrew Stewart

MICOM's SPECIAL INTEREST GROUPS

The CP/M Users Group of Melbourne (CPMUG) meets at 7:30 pm on the fourth Tuesday of each month. Meetings are held at Victoria College, Burwood Campus, in the Community Resources Centre (Building E). CPMUG can now provide public domain software on 65 different kinds of 5" disks, or standard 8" CP/M format.

Coming meetings: May 26, June 24

The Motorola Users Group Society (MUGS 68xx) meets at 7-30 PM on the second Tuesday of each month at the Box Hill Tech on the NW corner of Watts Street and Whitehorse road, Box Hill (room 13, 5th floor). NOTE NEW VENUE!
Contact Frank Crivelli on 434 3806 for further information.

Coming meetings: May 12, June 9

The Personal Computer Interest Group (PCIG - all aspects of MS-DOS/PC-DOS machines) meets for the time being after the monthly Micom meeting (see page two for venue details). Contact Andrew Moore on 596-1896 for details.

Coming meetings: May 16, June 20

The Otrona Users Group meets on the first Monday of each month, at the homes of its members. For information on the exact time and location of the next meeting, contact David Broadbent on 528 2792 AH.

Coming meetings: June 1, July 6

CONSUMABLES

3.5" SS Nashua disks (10)	34.56/box
3.5" DS Nashua disks (10)	45.36/box
5.25" SS/DD BASF disks (library case, 10)	19.87/box
5.25" DS/DD Nashua disks (10)	21.60/box
Plastic library cases	1.98 each
11"x9.5" 60gsm bond, margin perf's 2500 sheets	44.56/box
11"x9.5" 60gsm bond, margin perf's 500 pack	15-20
A4 70gsm bond margin perf's 2500 sheets	53.31/box
Rediform station - statements, invoices, etc	CALL
1.0"x3.5" plain labels (1000)	11.66/pack
1.5"x4.0" plain labels (1000)	21.00/pack
Coloured labels - blue, green, pink, yellow	CALL
Microline 80,82,83,92,93 printer ribbons	3.72 each
Microline 84 printer ribbons	8.35 each
Epson MX/FX/RX 80 printer ribbons	8.56 each
Epson MX/FX/RX 100 printer ribbons	15.48 each
NEC Spinwriter 3500 series multistrike ribbon	13.95 each
NEC Pinwriter P3 ribbon	20.80 each

HARDWARE

IBM PC compatible keyboards	120.00 each
NEC 1155 5.25" 80 track floppy drive	237.60 each
Mitsubishi 5.25" 40 track floppy drive	208.80 each
MicroMaster, 128K RAM, 6MHz HD64180	576.00 each
20MB 5.25" half height hard disk	734.40 each
Okidata-Microline printers	CALL

*** Quantity & sales tax exempt prices available
*** Bankcard and VISA welcome

Nemostar Computing (059) 786-133

MEMBERS' MESSAGES

Unclassified messages from MICOM members are FREE! If you would like to have a message placed in this section you can leave it on the bulletin board for the editor, come to a MICOM meeting and give it to the secretary or you could mail it.

FOR SALE

Toshiba P-1340 printer with parallel interface. This is a 24-pin letter-quality dot-matrix printer. It has two resident letter-quality fonts, Pica and proportional Courier. Printing speeds are:

high speed draft 10 cpi - 120 cps
letter-quality Pica - 47 cps
letter-quality proportional - 54 cps

It also features Qume emulation and an extra character set that can be accessed via printer cards capable of sending the 8th bit.

The new list price of similar products is in the $1200 range. My price is $800 which includes full documentation, two new ribbons and software for graphics dumps from Apple computers. The printer was purchased in July 1985.

If interested, please 'phone 836-6466

Roy McDonald

FOR SALE

I have the following Apple items for sale. I'm not putting prices on them, but will consider any reasonable offers!

Apple][+ Wildcard, joysticks (3 of them), numeric keypad for //e only, TG orig. paddles, Original Apple Parallel Card, Digicard serial card.

Just leave a message on the MICOM CBMS with any offers.

Martin Scerri

FOR SALE

The following items are intended mainly for use with Apple 2 computers.

Original Software

Megaworks, Pinpoint (inc. speller), Macroworks, Sideways, Copyplus, Essential III Backup, Apple Access, Terminapple, Viaterm, Netcom Videotex, an Introduction to Basic Programming and a Diagnostic Program. All software is complete with original manuals and, where supplied, example and training discs.

Miscellaneous Items

A reference book on Apple 2c and two giving introduction to C/PM, 5-1/4" diskette holders, used 5 1/4" floppy discs (mainly DS/DS).

Will sell all above items as one lot for $750 O.N.O. but am prepared to negotiate the sale of individual items. Leave message on the MICOM CBMS or contact Barry Clarke on 579-1928 AH or 537-2611 BH.

FOR SALE

Monochrome hi-res graphics card and monitor for IBM or comp. $200

Ring on 743-2447 AH or 391-1222 BH

Stephen Baker

HELP WANTED

I have a Big Board I computer with double-density drives. I have also built a 1 megabyte board from a kit - beautifully designed, beautifully constructed.

The trouble is - it doesn't work! I need some help. I have all the circuits etc. and am pretty sure there is not much wrong. But I don't have the equipment or skill to go any further.

I am willing to pay in money or equipment (I have a working ADM-1A terminal that I would be willing to part with) if anyone can help me get this board going.

Warwick Wright 850-8132

HELP WANTED

Are there any Turbo Pascal gurus out there who can help with a small problem that I've got?

I am using a defined file (which is a file made up of records defined in the TYPE section of the program) and I'm having problems with file positioning. When I retrieve (for instance) record number 2, update the data in the record, then try to write back to the same record, Turbo writes the data to the NEXT record!

So I tried using SEEK to reposition the file back to the record I wanted just before writing the record but the same thing happens.

Now, I've checked that the file pointer is pointing at record 2 (in the example above) right up to just before the write statement. It looks like Turbo is incrementing the file pointer, THEN writing the record. That doesn't seem right to me. I think it should write the record first, then increment the pointer.

Has anyone had this problem before? Can anyone suggest a solution? Writing to the record with a statement like:

```
seek(filename,recorpointer-1); write(filename,filerecord);
```

seems a bit messy to me - besides, what do you do when you're at the first record of the file?

If you've got any suggestions, leave a message on the MICOM CBMS.

Thanks - Bert Put

INFORMATION WANTED

I am a Spectrovideo user and am having great difficulty in obtaining PASCAL and Cobol language compiler disks. I would appreciate if anyone could help with information.

I am in need of CP/M programs in accounting and need CP/M BASIC, C, in fact anything and everything possible for the Spectravido 328. I have 62K RAM expansion, 80-column card, double-sided/ double-density disk drives.

Ring me on 700-3853 if you can help or write PO Box 222, Doveton.

Terry Rundell

INFORMATION WANTED

Could anyone suggest:

1 - a good book on Turbo-Prolog
2 - a good book on Prolog

If you can help please leave a message on the MICOM CBMS

Mirko Fluher

INFORMATION WANTED

Has anyone had any good/bad/indifferent reports to make re FTL Modula-2? I have purchased a compiler report and I am having trouble evaluating it (too late they cry)! My impression is generally favorable though the occaisional bug seems to bite.

Is anyone interested in sharing bits and pieces of code?

Is there enough interest to form yet another interest group?

Where should all of this be arranged?

Jim Ricketts

MAY/JUNE EVENTS

Sunday	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday
May 10 Amiga	11 Sega	12 MUGS 68xx (see note)	13	14	15	16 MICOM and PCIG
17	18	19	20	21	22	23 MICOM C'tee
24	25 Editor's Birthday	26 Cally's Birthday & CP/MUG	27	28	29	30
31	Jun 01 Otrona	02	03	04	05 Hitachi FIG	06 AUSOM
07 SCUA	08 Sega	09 MUGS 68xx	10	11	12	13
14 Amiga	15	16	17	18	19	20 MICOM and PCIG
22	23	24 CP/MUG	25	26	27	28 MICOM C'tee

Key to Event Calendar

AMIGA Amiga Users Group - Meets at Victoria College

AUSOM Apple Users Society Of Melbourne - Meets at Victoria College

CPMUG CP/M Users Group - Meets at Victoria College

MICOM Microcomputer Club Of Melbourne - Meets at Victoria College

MUGS 68xx Motorola 68xx Users Group - Meets Box Hill Tech, Watts Street

OTRONA Otrona Users Group - Meets at members' homes

PCIG Personal Computer Interest Group - meets at Vcitoria College for the time being

SCUA Sorcerer and CP/M Users of Australia - Meets at Victoria College

Where is Victoria College Burwood Campus?

New members and visitors sometimes have trouble locating our meeting place the first time. Its on the North side of Burwood Highway, Burwood, just East of Elgar Road. Coming from the City, turn left at the first set of traffic lights after Elgar Road. Follow the road around to the car parking areas, near the netball courts. Look around for Building E, and you've found us.

See page two of this newsletter for more information about the when and where of Micom meetings.

MICOM DISCOUNTERS

The following people are willing to give MICOM members discounts of various sorts. Please support them!

Daneva
64-66 Bay Road
Sandringham Vic 3191
10% discount.

de Forest Software
26 Station St.
Nunawading 3131
Peter DeForest was a MICOM member and is willing to give MICOM members discounts on software on presentation of their card.

Future Business Systems
6/105 Hawthorn Rd
Nth Caulfield 3161
Ph: 523 9211
Sol Millin is a MICOM member, and will give 10% discount to MICOM members who buy FBS products direct from him.

Gerardi Photographics
167 Elgin St.
Carlton 3053
Father of "Fulvius Maximus"!

JED Microprocessors
7/5-7 Chandler Rd
Boronia 3155
Ph: 762 3588
Ed Schoell is also a MICOM member, and will give a discount to bona-fide non-commercial purchasers of his STD-bus products on production of a MICOM membership card.

McGills Newsagency
Elizabeth St.
Melbourne 3000
Ph: 602 5566
Discount on computer books for MICOM members.

Technical Book and Magazine Co.
Swanston St
Melbourne 3000
10% on Books on computer related subjects.

THE MICROCOMPUTER CLUB OF MELBOURNE

May 1987 Newsletter

MICOM

MICOM Newsletter
Registered by Australia Post
Publication No. VBH 3601
If undeliverable, return to:
PO Box 60, Canterbury, 3126

POSTAGE PAID

BORONIA 3155